# Ocean Modeling on 1152 Cray T3E processors

Alan J. Wallcraft
Naval Research Laboratory, Stennis Space Center, MS 39529-5004

## Abstract

A $1/32^\circ$ (3.5 km at mid-latitudes) global ocean model was run for 3.5 model years on 1152 T3E processors. The highest resolution achieved globally outside DoD is 20 km at mid-latitudes by the Los Alamos National Laboratory (LANL). The lessons learned from running on a large number of processors will be discussed. Some of these are application specific, but others are relevant to many large scalable projects.

HPCMP now has 1,904 T3E processors but these are spread over four sites, with the largest single machine, at NAVO MSRC, containing 782 application processors. Recently, Cray Research assembled a single machine with 1,488 application processors on their factory floor in Chippewa Falls. This was primarily for benchmarking, but they allowed us to run a ”production” $1/32^\circ$ global ocean model on the machine. This is the highest resolution ever used globally for an ocean model, and it gave us a head start at a resolution that we are scheduled to use globally under DoD Challenge in FY00.

In dedicated mode, we used 1152 processors and each model month took 3.1 wall hrs. Including initialization, but excluding debugging time, the total 3.5 year simulation took 6 days of dedicated machine time. Counting only useful operations, we are running at about 100 Gflops on 1152 nodes. Of more importance is scalability, and our measured speedup for a practical run with all the usual I/O was 1.81x between 384 and 768 nodes and 1.58x between 576 and 1152 nodes. We expect to see super-linear speedup (above 2x) between 768 and 1536 nodes, because the global land-sea geometry is particularly well suited to 1536 nodes. However, the machine was just slightly too small to test this.

We have demonstrated that large node count machines are viable platforms for ocean modeling. Our goal is to make $1/32^\circ$ global simulations routine over the next few years, and it is now clear that one path to this goal is for shared-resource machines, or machine clusters, to be much larger - two to four thousand nodes each.

## Introduction

NRL's coordinated 6.1-6.4 effort on the “Grand Challenge” problem of eddy-resolving global and basin-scale ocean modeling and prediction has as its long term objectives to simulate, understand, nowcast and forecast the global ocean circulation, and to increase the capability to model it. In particular, we are working towards delivering by about FY05 a data assimilative $1/32^\circ$ global ocean nowcast/forecast system to Fleet Numerical Meteorology and Oceanography Center (FNMOC) for operational evaluation. We have already delivered a $1/4^\circ$ global product, and will deliver other basin and global systems with increasing resolution between now and FY05. Our goal is $1/32^\circ$ because we have considerable evidence that this resolution is required to well resolve the oceanic eddy space scale. Our DoD Challenge paper [1] provides more details on this subject. The NRL Layered Ocean Model (NLOM) [2] has a factor of 10s to 100s advantage in computer time requirements over other global and basin-scale ocean models. Even so, $1/32^\circ$ global NLOM is at the very edge of computational feasibility on existing HPC systems. If we are to deliver an operational system with this resolution within 5-6 years, running models of this size must become routine within the next 2-3 years.

Each doubling of horizontal resolution halves the time step and therefore requires eight times as much computer time (NLOM does not require an increase in vertical resolution). Moore's Law implies a doubling of processor speed every eighteen months, so eight times speedup takes about five years. The only certain way to improve on this time scale is to make efficient use of more processors as the problem size increases. Given a limited budget, scaling to more processors will only help if the per processor cost of large state of the art machines goes down over time (while still following Moore's Law single processor performance). There is some prospect that this will happen, given the large difference in total system price per processor between MPP's and smaller commodity machines today.

HPCMP now has 1,904 T3E processors but these are spread over four sites, with the largest single machine, at NAVO MSRC, containing 782 application processors. These are shared machines and, outside of DoD Challenge, the maximum number of processors routinely available to a single job is about 128 at NAVO and about 256 at CEWES. CEWES has only 520 T3E nodes but its batch configuration is tuned for large jobs. Recently, Cray Research assembled a single machine with 1,488 application processors on their factory floor in Chippewa Falls. This was primarily for benchmarking, but they allowed us to run a ”production” $1/32^\circ$ (3.5km) global ocean model on the machine. This is the highest resolution ever used globally for an ocean model, and it gave us a head start at a resolution that we were scheduled to use globally under DoD Challenge in FY00. The highest resolution achieved globally outside DoD is 20 km at mid-latitudes by the Los Alamos National Laboratory (LANL).

## Model Characteristics

Almost all ocean general circulation models (OGCM's), including NLOM, solve the hydrostatic Bousinessq primitive equations using 2nd order finite differences in the horizontal. They primarily differ in how they treat the vertical dimension and in their time stepping scheme (how they avoid the small time step implied by gravity waves). NLOM uses the most efficient proven scheme in both areas. It uses only 5-6 Lagrangian layers in the vertical and a semi-implicit time scheme that treats all gravity waves implicitly. There are no “free lunches”, and NLOM gains its 10-100x advantage in computer time in part by giving up the ability to perform some kinds of climate studies and to model coastal regions. All OGCM's have much larger horizontal than vertical dimensions, and they therefore all scale well using 2-D domain decomposition. Explicit time step models potentially scale better than implicit models, but it seems to be a general rule for CWO models that techniques that improve efficiency are a “win” even if they tend to reduce scalability. In the case of NLOM, the semi-implicit time scheme involves a direct 2-D Helmholtz's equation solver that requires transposing from a 2-D to a 1-D domain decomposition, and therefore potentially reduces overall scalability. In general, scalability of NLOM [3] [4] is excellent on current scalable systems (using 64-256 nodes per job) because the 2-D arrays are so large.

Our $1/32°$ global (near-global, 72S-65N) grid is 8192 by 4608 by 6. The largest practical number of processors for a problem this size using NLOM is 1536. By discarding sub-domains that are completely over land, we can configure for 2048 virtual processors (64 by 32) but discard 25% to get down to 1536 actual processors. The 2-D sub-domain size is then 128 by 144, which still provides enough work per sub-domain to amortize communication overhead. However, the 1-D sub-domain size is 8192 by 3 and for some computations it is the small second dimension that controls the amount of work.

NLOM's direct 2-D Helmholtz's equation solver uses the Capacitance Matrix Technique [5], which involves a dense linear system across all coastline grid points. For the $1/32°$ global region this is a huge matrix, 83,000 by 83,000. Two of these matrices are required, and the (time invariant) linear system must be solved at every time step. This is perhaps the largest dense set of linear equations ever involved in a time dependent problem. The memory required for these matrices sets a lower limit on the number of processors. Using NAVO's new 512 MB nodes we can run on 168 processors, but more than 300 of the standard 256 MB T3E nodes would be required.

Disk and archive requirements are also large. The two huge matrices are not recalculated for each run, but are instead read in from disk, and each require 26 GB of scratch disk space. A typical complete experiment simulates 15 to 25 years, but each individual run covers one month. Writing a “history record” (that allows restarts and is also used diagnostically) 10 times a month creates a 25 GB file. We would like to write once a day, but that would create a 75 GB file per month. Archiving one experiment therefore requires 5 to 20 TB. Transferring multiple 25 GB files between machines, even within a single computer center, can place a significant load on the network.

## Using Many Processors

NLOM performs the majority of communication via calls to an application specific API. This greatly simplifies portability (since the details of communication are hidden by the API), but it means that computation and communication are distinct phases with little overlap between them. Communication latency cannot therefore be effectively overlapped with computation, and so low latency becomes increasingly important as the number of processors is increased for a fixed problem size. In principle, a large scalable problem on many processors should perform similarly to a small problem on a few processors. This is in fact the most common definition of a “scalable” code. Again, for NLOM, low latency becomes increasingly important as the problem size is reduced for a fixed number of processors. Figure 1 shows a low level communication benchmark on 16 processors. It is simulating a nearest neighbor halo exchange for a square sub-domain of size N by N, with N varying from 2 to 1024. At large N bandwidth dominates, but as N gets smaller latency becomes increasingly important. Across a variety of machine types, MPI [6] is exhibiting much higher latency than any one of several techniques using direct memory access. The T3E using SHMEM [7] provides both the lowest latency and the highest bandwidth of all current HPC systems. On the T3E it is normal to synchronize SHMEM transfers using a global barrier, which is supported in hardware. Figure 2 shows that on machines without a hardware barrier, local synchronization is much more efficient than a global barrier. Global barrier's become more expensive as the number of processors increases, and even on the T3E local synchronization is faster on more than about 256 processors. Therefore local synchronization options added to NLOM for SHMEM on the Origin 2000 are equally applicable to the T3E when many processors are used. Additional local synchronization support was required in a pipelined tridiagonal solver that is particularly sensitive to the 1-D decomposition sub-domain aspect ratio.

An important design goal of NLOM is bit for bit reproducibility no matter how many nodes are used, i.e. identical answers on M and N processors for all likely M and N. This is a common (but not universal) requirement across the CWO community. Testing the correctness of highly non-linear models is very time consuming, and bit for bit reproducibility allows the testing on N processors to apply equally to M processors. The fastest methods for global operations (e.g. global sum) are not bit for bit, and many standard libraries allow such methods. For example, `MPI_REDUCE` is not required to return the same answer if called twice with the same inputs even on the same number of processors. Global operations become relatively more expensive on large node counts. Even on small numbers of nodes they can be expensive using high latency MPI point to point calls. NLOM's approach to global sums, optimized for MPI, is to split each row sum into a fixed number of pieces (an integer multiple of the number of nodes per row) which are performed independently. These are then combined in a fixed order, but each row is independent. Finally the row sums are combined in a fixed order (not scalable, but inexpensive). This approach is bit for bit reproducible whenever the same number of pieces are used per row. Although designed for MPI on about 64 processors, this also works well using SHMEM on about 1000 processors. In general, the effort expended on optimizing NLOM for the IBM SP with MPI has significantly improved Cray T3E with SHMEM performance for large numbers of nodes.

Each one month run must read in two 26 GB matrices and a 2.5 GB restart record, and write out 10 to 30 2.5 GB history records (25 GB to 75 GB total). NLOM implements parallel I/O, but this does not typically scale beyond a few 10's of processors involved in the I/O. The history record writes could be completely overlapped with computation if performed asynchronously using a 2.5 GB distributed memory buffer. However, there is no portable way to implement this in Fortran. Therefore both the reads and the writes use synchronous I/O and all processors are otherwise idle while I/O occurs. The T3E has a relatively high performance I/O sub-system, but its speed depends on what kind of disks are used and how they are configured. On 1152 processors the total I/O wall time was 25 minutes per model month (13% of total wall time). On 168 NAVO T3E processors this is reduced to 8 minutes (1% of total wall time).

## Huge Linear Systems

The 83,000 by 83,000 dense linear systems caused the majority of the problems encountered in getting the simulation running. NLOM's original approach was to form the inverse of the matrix once and then perform a matrix vector multiply to solve the system every time step. Forming the inverse explicitly is not usually recommended for such systems, but it has the advantage that the solve phase is trivially bit for bit reproducible and optimally scalable. Normally the $O(N^3)$ cost of inverting the system dominates the $O(N^2)$ cost of each solve, but here the total cost of all solves also grows as $O(N^3)$ and dominates over the inversion phase. The inversion cost was so cheap for most regions encountered that a trivial parallelization of a LINPACK routine was “good enough”, even though much better methods now exist. However, starting with $1/16^\circ$ global simulations (first run under DoD Challenge in FY97) it became apparent that even if the relative cost of the inversion phase was still low, its absolute cost was becoming an issue. Standard packages, such as ScaLAPACK, are not bit for bit reproducible and therefore cannot be used for the per time step solve phase. The matrix inverse is only formed once, so bit for bit issues do not apply and ScaLAPACK could be used for this phase. However in FY97 ScaLAPACK was not available for 32-bit REAL on the T3E, and minor optimizations to the existing code allowed it to form each inverse in "only" seven hours on 224 processors. When this method was applied to the $1/32^\circ$ case on 1152 processors it took 20 hours for each inverse. Since the T3E was an experimental configuration used primarily for relatively short benchmarks, it was difficult to get two 20-hour time slots to make the inverses and debugging required several re-creations of the inverses. The normal approach to solving dense linear systems is to form its LU decomposition, where L and U are triangular systems. On serial machines it is trivial to solve triangular systems, but on scalable machines these solves do not readily parallelize. An alternative is to form the inverse of L and of U, which can then be used to solve the original system using matrix-vector multiplies. With this approach, each solve is almost as fast as using the original inverse directly but forming the inverses is much faster. Now the initial phase takes only 5 hours per matrix. On 168 T3E nodes it still takes 24 hours per matrix, so further optimization would be helpful, but the overall cost (10,000 processor hours) is now trivial relative to the overall cost of a complete simulation.

With a matrix this large the accuracy of the linear system solution is obviously a concern, particularly since NLOM uses 32-bit REALs. The magnitude of each matrix element depends on the distance between the two associated coastline points, and falls of quickly as this distance increases. The condition number of the matrix is small and proportional to the log of its dimension, i.e. grows only slowly with increased grid resolution. Finally, the system solved is the correction to the previous time step's result. In practice, 32-bit REALs have been sufficient at least up to the current $1/32^\circ$ global case.

## Performance

The Chippewa Falls machine was run in dedicated single user mode, so there was no advantage to discarding virtual processors over land. The majority of runs were made using 1152 processors and each model month took 3.1 wall hrs. Counting only useful operations, we are running at about 100 Gflops on 1152 nodes. Of more importance is scalability, and our measured speedup for a practical run with all the usual I/O was 1.81x between 384 and 768 nodes and 1.58x between 576 and 1152 nodes. We expect to see super-linear speedup (above 2x) between 768 and 1536 nodes, because (as explained above) the global land-sea geometry is particularly well suited to 1536 nodes. However, the machine was just slightly too small to test this. Note that I/O time is constant from 384 to 1152 nodes and accounts for much of the loss of scalability.

Including initialization, but excluding debugging time, the total 3.5 year simulation took 6 days of dedicated machine time. We are in the process of extending the simulation a further 3 years at NAVO MSRC under FY99 DoD Challenge, and our FY00 proposal includes 15 model years of a more realistic "thermodynamic" version of the same model.

## Conclusions

We have demonstrated that the $1/32^\circ$ global model can run efficiently on up to 768 processors, and perhaps on twice that number. Our goal is to make $1/32^\circ$ global simulations routine over the next few years, and it is now clear that one path to this goal is for shared-resource machines, or machine clusters, to be much larger. In order for 400-800 processors to be routinely available to a non-Challenge job, such a machine or cluster would need to contain two to four thousand nodes.

**Acknowledgments**

*This is a contribution to the 6.2 Global Ocean Prediction System Modeling Task. Sponsored by the Office of Naval Research under Program Element 62435N. Also to the Common HPC Software Support Initiative project Scalable Ocean Models with Domain Decomposition and Parallel Model Components. Sponsored by the DoD High Performance Computing Modernization Office. Cray Research provided the computer time for the* $1/32°$ *global simulation on a 1,488 application processor Cray T3E at Chippewa Falls WI. Many thanks to Ilene Carpenter of Cray Research for her assistance in making the simulation possible.*

# References

[1] Wallcraft A.J, H.E. Hurlburt, R.C. Rhodes, J.F. Shriver (1999). Global and Basin-Scale Ocean Modeling and Prediction. *DoD HPC Users Group Meeting*, Monterey, CA, June 7-10, 1999.

[2] Wallcraft A.J (1991). The NRL Layered Ocean Model users guide. *NOARL Report* **35**. Naval Research Laboratory, Stennis Space Center, MS, 21 pp.
**http://www7300.nrlssc.navy.mil/html/images/users_guide.ps.gz**

[3] Wallcraft A.J & D.R. Moore (1996). A Scalable Implementation of the NRL Layered Ocean Model. *NRL Contractor Report* **CR 7323–96-0005** Naval Research Laboratory, Stennis Space Center, MS.

[4] Wallcraft A.J & D.R. Moore (1997). The NRL Layered Ocean Model. *Parallel Computing* **23**, pp 2227-2242.

[5] Buzbee, B.L. F.W. Dorr, L.A. George, G.H. Golub (1971). The direct solution of the discrete Poisson equation on irregular regions. *SIAM J. Num. Anal.*, **8**, pp 722-736.

[6] Snir M., S.W. Otto, S. Huss-Lederman, D.W. Walker, J. Dongarra (1996). *MPI: the complete reference*. MIT Press.

[7] Cray Research Inc. (1996) Application Programmer's Library Reference Manual. *Cray Research SR-2165*

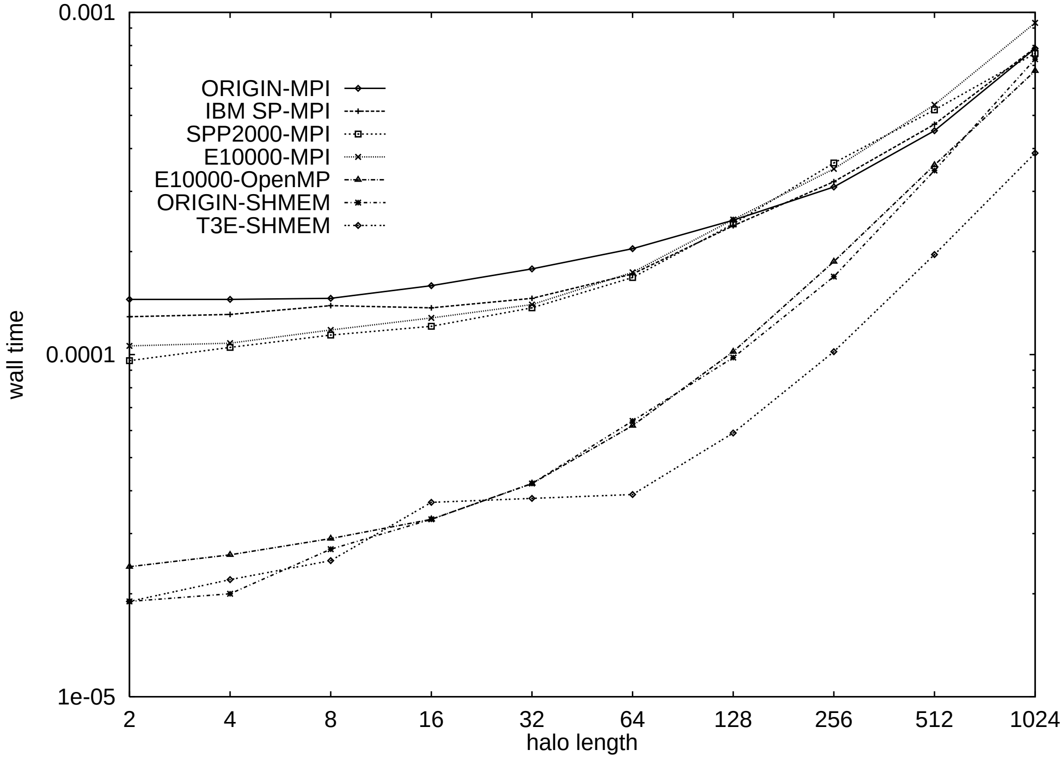

Figure 1: Best HALO times on 16 processors

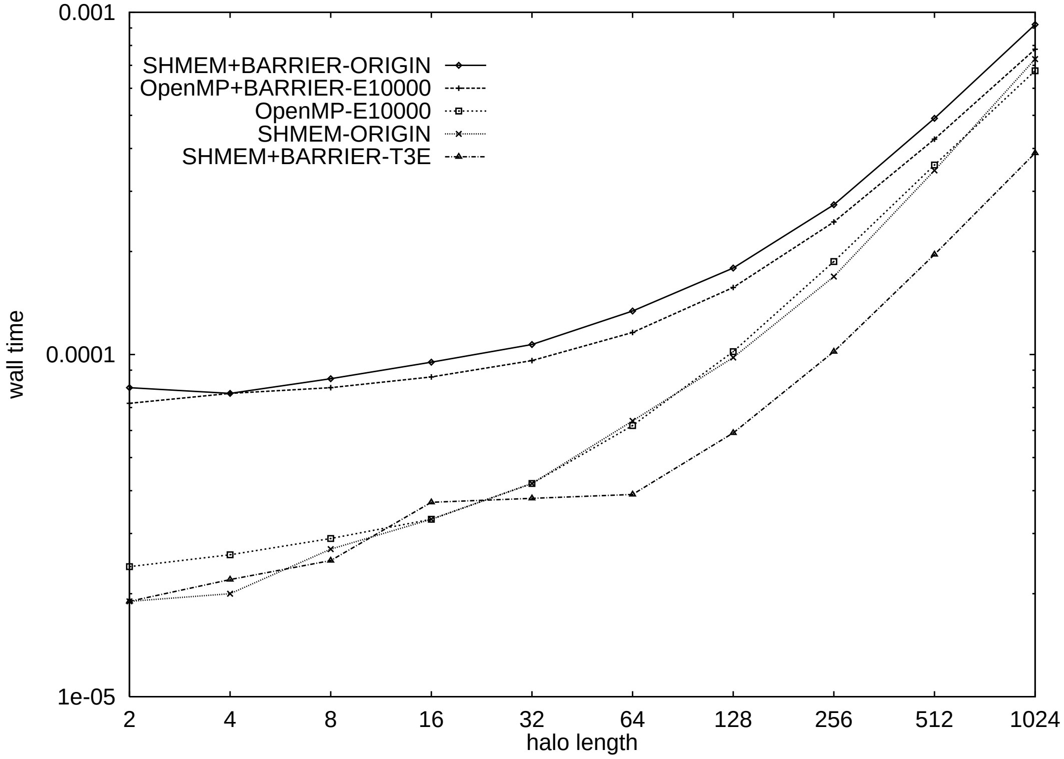

Figure 2: Shared memory HALO times on 16 processors